J2EE Web Services

- J2EE web services **build on** existing J2EE technologies
- Servlets, JSPs can be called as web services
  - This is because they already can be called via HTTP
  - Now you're just happening to send XML instead of HTML
J2EE Web Services

- This was somewhat cumbersome
  - You needed to write your own logic to perform web services operations
- Now it's better
  - You have some helper APIs to help you do SOAP, UDDI, WSDL, ebXML
  - After all, you don't want to write code that goes down to the low-level of XML/HTTP
  - Those helper APIs are JAXR, JAXB, JAXM, JAX/RPC, and jWSDL
J2EE Web Services

- Each J2EE Web Services API helps you perform a web services operation
  - JAXR: UDDI
  - JAXM: ebXML
  - JAX/RPC: SOAP
  - jWSDL: WSDL
  - JAXB: Transforming Java objects to and from XML
J2EE Web Services Iteration #2

1: Register Service in UDDI Registry

2: Locate Service and/or Service Descriptions

3: SOAP (XML/HTTP) possibly with ebXML messages

4: Process XML-based RPC (SOAP) request

- Business Partner or Other System
- UDDI Service
- Firewall
- Container
- Servlets
- JAX/RPC
- JAXM
- JAXB
- JAXR
J2EE Web Services

So the great thing with J2EE is:

- You write a reusable enterprise app with EJB
- You can then expose that app as a
  - Web site (Servlets/JSP)
  - Traditional GUI application (Application or Applet)
  - Web service (Servlets, JAX APIs, EJB 2.1)